

Special Copyright Notice

© 1991 by the American Institute of Aeronautics and Astronautics. All rights reserved.

Guide

Guide for Implementing Software Development Files Conforming to DoD-STD-2167A



ANSI / AIAA
G-009-1991

American National Standard

Guide for Implementing Software Development Files Conforming to Dod-STD-2167A

Sponsor:

American Institute of Aeronautics and Astronautics

Approved: May 20, 1991

American National Standards Institute, Inc.

Abstract

The DoD Standard, DoD-STD-2167A, establishes uniform requirements for software development that are applicable throughout the system life cycle, and it is used by several Federal Agencies. That standard has been recognized to be applicable to software developments in the larger aerospace community. This AIAA Guide facilitates the computerized implementation of software development file requirements where a large part of the information is provided through the software engineering environment.

American National Standard

Approval of an American National Standard requires verification by ANSI that the requirements for due process, consensus, and other criteria have been met by the standards developer.

Consensus is established when, in the judgement of the ANSI Board of Standards Review, substantial agreement has been reached by directly and materially affected interests. Substantial agreement means much more than a simple majority, but not necessarily unanimity. Consensus requires that all views and objections be considered, and that a concerted effort be made toward their resolution.

The use of American National Standards is completely voluntary; their existence does not in any respect preclude anyone, whether he has approved the standards or not, from manufacturing, marketing, purchasing, or using products, processes, or procedures not conforming to the standards.

The American National Standards Institute does not develop standards and will in no circumstances give an interpretation of any American National Standard. Moreover, no person shall have the right or authority to issue an interpretation of an American National Standard in the name of the American National Standards Institute. Requests for interpretations should be addressed to the secretariat or sponsor whose name appears on the title page of this standard.

CAUTION NOTICE: This American National Standard may be revised or withdrawn at any time. The procedures of the American National Standards Institute require that action be taken to affirm, revise, or withdraw this standard no later than five years from the date of approval. Purchasers of American National Standards may receive current information on all standards by calling or writing the American National Standards Institute.

Guide for implementing software development files conforming to DoD
-Std-2167A : American National standard / sponsor, American
Institute of Aeronautics and Astronautics.

p. cm.

"Approved, American National Standards Institute, Inc."

ANSI/AIAA G-009-1990."

Includes bibliographical references.

ISBN 0-930403-81-9 (pbk.)

1. Computer software --Development--Standards--United States.

I. American Institute of Aeronautics and Astronautics. II. American
American National Standards Institute.

QA76.76.D47G85 1990

005.1'021873--DC20

90-19729

CIP

Published by

American Institute of Aeronautics and Astronautics
370 L'Enfant Promenade, SW, Washington, DC 20024

Copyright © 1991 American Institute of Aeronautics and Astronautics
All rights reserved

No part of this publication may be reproduced in any form, in an electronic
retrieval system or otherwise, without prior written permission of the publisher.

Printed in the United States of America

CONTENTS

	Foreword	v
1.0	Introduction	1
1.1	Background	1
1.2	Purpose and scope	1
2.0	Applicable documents.....	1
3.0	A brief review of SDFs	1
3.1	Definition	1
3.2	Discussion	1
4.0	SDF requirements	2
4.1	DoD-Std-2167A requirements	2
4.2	Developer selected requirements	4
5.0	Analysis of DoD-Std-2167A SDF requirements	5
5.1	The total requirements	5
5.2	Organization of the requirements	5
5.3	SDF data items	5
5.4	Timing during the life cycle	5
5.5	Critique of DoD-Std-2167A SDF requirements	5
6.0	Recommended process for SDFs.....	7
6.1	The SDF process	7
6.1.1	Step 1 Establish project SDF requirements	7
6.1.2	Step 2 Tailor DoD-Std-2167A SDF requirements in the contract statement -ofwork	7
6.1.3	Step 3 Analyze contract SOW for SDF requirements	10
6.1.4	Step 4 Add developer requirements	10
6.1.5	Step 5 Select SDF approach using design issues and trades	10
6.1.6	Step 6 Specify SDF data items and procedures in Para. 4.2.2 of SDP	10
6.1.7	Step 7 See specifications and development procedures	10
6.1.8	Step 8 Project training	10
6.1.9	Step 9 Environment implementation	11
7.0	Design issues and trades	11
7.1	General issues for developing the SDFs.....	11
7.2	SDF data flow analysis and design	11
7.3	Organizational structure and roles	11
7.4	Storage of SDF information	12
7.5	Use of a file directory structure	12
7.6	Limitations on file directory structure	12
7.7	Use of a database management system (DBMS)	12
7.8	Special SDF tools	12
7.9	Configuration Management of SDF Information	13
7.10	Security	13
8.0	References	13
9.0	Acronyms.....	13

List of Tables and Figures

Table	5-1	Requirements Expansion	6
Figure	5-1	SDF Requirements	6
Figure	5-2	Timing of SDF Requirements during the DoD-Std-2167A Life Cycle	8
Figure	6-1	SDF Process	9

Foreword

This *Guide for Implementing Software Development Files Conforming to DoD-Std-2167A* has been sponsored by the American Institute of Aeronautics and Astronautics (AIAA) as part of its Standards Program.

In DoD-Std-2167A, the precise definition of SDFs is deferred to the contracting agency and the contractor. The content, style, and format of SDFs are not specified in Data Item Descriptions as other deliverable documents under DoD-Std-2167A. Therefore, the contents of SDFs and whether they are in hard copy or computerized form may differ widely between different defense system software developers.

This AIAA Guide is based on an analysis of the requirements in DoD-Std-2167A for Software Development Files and presents a generalized process for implementation. Important design issues and trades are discussed to aid the practical implementation by individual organizations. Although it is specifically written to guide the user to be compliant with DoD-Std-2167A, similar requirements occur in other government agencies and within the larger aerospace industry.

The AIAA Standards Procedures provide that all approved Standards, Recommended Practices, and Guides are advisory only. Their use by anyone engaged in industry or trade is entirely voluntary. There is no agreement to adhere to any AIAA standards publication and no commitment to conform or be guided by any standards report. In formulating, revising, and approving

standards publications, the Committees on Standards will not consider patents that may apply to the subject matter. Prospective users of the publications are responsible for protecting themselves against liability for infringement patents or copyrights or both.

AIAA Software Systems Committee on Standards

At the time of preparation, the following individuals were member of the AIAA Software Systems Committee on Standards:

H. G. Stuebing Chairman (Naval Air Development Center)
 John Brackett (College of Engineering, Boston University)
 Edward Comer (Software Productivity Solutions)
 Alvin Fletcher (TRW, Inc)
 Beverly Kitaoka (Science Applications International Corporation)
 Frank LaMonica (Rome Air Development Center)
 Brian Larman (Jet Propulsion Laboratory)
 Robert Martin (Raytheon Company)
 Daniel McNicholl (McDonnell-Douglas)
 Randall Scott (Software Productivity Consortium)
 Andrea Federoff Sebera (Space Systems / Loral)
 William Wishon (Hughes Aircraft Company)

This document was approved by the Software Systems Committee on Standards in April 1990.

The AIAA Standards Technical Council (William W. Vaughan, Chairman) approved the document in November 1990.

1.0 Introduction

1.1 Background

DoD-Std-2167A *Defense System Software Development* defines the term "software development files (SDFs)" and specifies that contractors shall establish, implement, and maintain SDFs for the duration of the contract. The SDFs are to be made available for contracting agency review on request.

In DoD-Std-2167A the precise definition of SDFs is deferred to the contracting agency and the contractor. The content, style, and format of SDFs are not specified in Data Item Descriptions as other DoD-Std-2167A deliverable documents. Therefore, the contents of SDFs and whether they are in hard copy or computerized form may differ widely between different defense system software developments.

DoD-Std-2167A also encourages the automatic production of all deliverable data. Additionally, DoD-Std-2167A requires contractors to establish both a software engineering environment (SEE) and a software development library (SDL). These requirements reflect the general industry trend toward higher degrees of automation in all aspects of software development, including all software and associated documentation. When a SEE and SDL are delivered to the government for post deployment software support, any computerized SDFs implicitly become deliverable items.

1.2 Purpose and scope

The purpose of this AIAA Guide is to discuss the issues and trade-offs that impact the practical implementation of SDFs that conform to DoD-Std-2167A requirements. Although implementing SDFs in purely manual hard copy form is not precluded, this Guide is oriented toward the computerized implementation of SDFs where a large part of the required information is automatically provided by the SEE. The SDFs then become a part of the SDL.

This Guide is intended to be widely applicable in software developments throughout the larger aerospace community. Although it is specifically written to guide the user to be compliant with DoD-Std-2167A, similar requirements for SDFs occur in other government agencies and within industry.

2.0 Applicable documents

2.1 DoD-Std-2167A *Defense System Software Development*; 29 Feb 1988

2.2 DoD-Hbdk-287 A Tailoring Guide for DoD-Std-2167A, *Defense System Software Development*; (Draft) 14 Nov 1988

2.3 IEEE Std 729-1983 *Glossary of Software Engineering Terminology*

3.0 A brief review of SDFs

3.1 Definition

Software Development File (SDF) A repository for a collection of material pertinent to the development or support of software. Contents typically include (either directly or by reference) design considerations and constraints, design documentation and data, schedule and status information, test requirements, test cases, test procedures, and test results.¹

3.2 Discussion

The idea of an SDF can be traced to Unit Development Folders (UDFs), which have been also called Software Development Folders and Unit Development Notebooks. The concept originated about 1975 and was intended to provide better control and visibility of the software development process. Most software organizations had been using project notebooks to collect the large amount of detailed data generated during a software de-

¹DoD-Std-2167A Para. 3.26 page 6

velopment; however, these notebooks were usually informal, unstructured documents with varying levels of quality and completeness.

By formalizing the structure and content of the notebooks, the UDFs became both a working document and an audited controlled document. This gave management visibility into the progress of each software unit in the project. As the name "folder" implies, the UDFs were created and maintained in hard copy form.

A UDF is usually started by the end of the preliminary design phase when specific requirements are firmly allocated to the software. A software engineer is assigned custodian of the UDF and is responsible to define the intermediary milestones and to complete each section as the development progresses. A review authority frequently checks the UDF and approves the accomplishment of milestones; typically, the start and completion of the work for each section is recorded on a one page cover sheet.

The following is a typical organization of a UDF:

1. Cover Sheet and Schedule
2. Requirements
3. Design Description
4. Unit Test Plan
5. Unit Code
6. Test Case Results
7. Notes
8. Reviewer's Comments

An SDF is the same concept as a UDF; the term "file" reflects the more modern trend of recording the information in a computer file. [1,2,3,4]

4.0 SDF Requirements

4.1 DoD-Std-2167A Requirements

R1 Para. 4.2.4 Non-developmental software, page 14

"The SDFs for NDS (non-developmental software) need not contain the design considerations, constraints, or data."

R2 Para. 4.2.9 Software development files, page 14

"The contractor shall document the development of each Computer Software Unit (CSU), Computer Software Component (CSC), and Computer Software Configuration Item (CSCI) in the SDFs. The contractor shall establish a separate SDF for each CSU or a logically related group of CSUs, each CSC or logically related group of CSCs, and each CSCI. The contractor shall document and implement procedures for establishing and maintaining SDFs. The contractor shall maintain the SDFs for the duration of the contract. The SDFs shall be made available for contracting agency review upon request. SDFs may be generated, maintained, and controlled by automatic means. To reduce duplication, SDFs should not contain information provided in other documents or SDFs. The set of SDFs shall include (directly or by reference) the following information:

- a. Design considerations and constraints
- b. Design documentation and data
- c. Schedule and status information
- d. Test requirements and responsibilities
- e. Test cases, procedures, and results. "

R3 Para. 5.3.2.4, page 23

During the preliminary design phase, "The contractor shall establish test requirements for conducting CSC integration and testing. The contractor's CSC integration and testing shall include stressing the software at the limits of its specified requirements. The contractor shall record the test requirements (directly or by reference) in the CSC software development files."

R4 Para. 5.4.2.4 page 25

During the detailed design phase, "The contractor shall establish test responsibilities, test cases (in terms of inputs, expected results, and evaluation criteria), and schedules for CSC integration and testing. The contractor

shall record this information (directly or by reference) in the CSC software development files."

R5 Para. 5.4.2.5, page 25

During the detailed design phase, "The contractor shall establish test requirements, test responsibilities, test cases (in terms of inputs, expected results, and evaluation criteria), and schedules for testing all CSUs (Computer Software Unit). The contractor's CSU testing shall include stressing the software at the limits of its specified requirements. The contractor shall record this information (directly or by reference) in the CSU software development files."

R6 Para. 5.4.4.e, page 25

"The contractor shall perform evaluations of the products identified below, using the evaluation criteria specified in Figure 7. The contractor shall present a summary of the evaluation results at the critical design review(s)."

"E. A specified percentage of the set of CSU and CSC software development files (SDFs). The specified percentage shall be as identified in the Software Development Plan (SDP)."

R7 Figure 7, page 26

The evaluation criteria of Figure 7 state that the contents of CSU and CSC SDFs shall have (1) internal consistency, (2) understandability, (3) traceability of CSU SDFs to CSC SDFs, and (4) consistency. Specific definitions of terms of the evaluation criteria are in Appendix D of DoD-Std-2167A.

R8 Para. 5.5.2.1, page 27

During coding and CSU testing, "The contractor shall develop test procedures for conducting each CSU test. The contractor shall record these procedures in the corresponding CSU software development files (SDFs)."

R9 Para. 5.5.2.2, page 27

During coding and CSU testing, "The contractor shall code and test each CSU ensuring

that the algorithms and logic employed by each CSU are correct and that the CSU satisfies its specified requirements. The contractor shall record the test results of all CSU testing in the corresponding CSU SDFs."

R10 Para. 5.5.2.3, page 27

During coding and CSU testing, "The contractor shall make all necessary revisions to the design documentation and code, perform all necessary retesting, and shall update the SDFs of all CSUs that undergo design or coding changes based on CSU tests."

R11 Para. 5.5.2.4, page 27

During coding and CSU testing, "The contractor shall develop test procedures for conducting each CSC test. The contractor shall record these procedures in the CSC SDFs."

R12 Para. 5.5.4.d, page 27

"The contractor shall perform evaluations of the products identified below, using the evaluation criteria specified in Figure 8.

d. A specified percentage of the set of updated software development files (SDFs)."

R13 Figure 8, page 28

The evaluation criteria of Figure 8 state that the contents of CSU and CSC SDFs shall have (1) internal consistency, (2) understandability, (3) traceability of CSU SDFs to CSC SDFs, and (4) consistency. Specific definitions of terms of the evaluation criteria are in Appendix D of DoD-Std-2167A.

R14 Para. 5.6.2.2, page 29

During CSC integration and testing, "The contractor shall record the test results of all CSC integration and testing in the corresponding CSC software development files (SDFs)."

R15 Para. 5.6.2.3, page 29

During CSC integration and testing, "The

contractor shall make all necessary revisions to the design documentation and code, perform all necessary retesting, and shall update the software development files (SDFs) of all CSUs, CSCs, and CSCIs that undergo design or coding changes based on CSU tests."

R16 Para. 5.6.3.2, page 29

Before conducting formal Qualification Testing (FQT), "The contractor shall record the test results of this activity in the corresponding CSCI software development files (SDFs) and shall update the Software Test Description (STD) as appropriate."

R17 Para. 5.6.4.a, page 29

"The contractor shall perform evaluations of the products identified below, using the evaluation criteria specified in Figure 9. The contractor shall present a summary of the evaluation results at the Test Readiness Review.

"A. The test results recorded in the software development files (SDFs)"

R18 Para. 5.6.4.d, page 29

"The contractor shall perform evaluations of the products identified below, using the evaluation criteria specified in Figure 9. The contractor shall present a summary of the evaluation results at the Test Readiness Review."

"D. A specified percentage of the updated software development files (SDFs)."

R19 Figure 9, page 30

The evaluation criteria of figure 9 states that the Contents of updated SDFs shall have (1) internal consistency, (2) understandability, (3) consistency with updated code and SDD, and (4) adequacy of updated test results. Specific definitions of terms of the evaluation criteria are in Appendix D of DoD-Std-2167A.

R20 Para. 5.7.2.1, page 31

During CSCI testing, "The contractor shall

make necessary revisions to the Software Design Document(s) (SDDs) and code, conduct all necessary retesting, and update the Software Development Files (SDFs) of all CSUs, CSCs, and CSCIs that undergo design or coding changes based on the results of formal qualification testing."

**R21 From DI-MCCR-80030A,
Software Development Plan (SDP)**

10.2.6.2.2 Software Development Files, page 8 "This subparagraph shall be numbered 4.2.2 and shall define the contractor's plans, including the responsible organization(s), for the creation and maintenance of Software Development Files (SDFs). This subparagraph shall define the format and contents of the SDFs and describe the procedures for maintaining SDFs."

4.2 Developer selected requirements

Many organizations have internal processes and procedures for the development of software. These organizations may require that their personnel or subcontractors record additional information in the SDF beyond the items required by DoD-Std-2167A.

The following list shows items that a developer may include in a SDF:

- Cover sheet and schedule,
- Action items,
- Block diagrams,
- Log of changes to the SDF,
- Error status and history,
- Limitations and restrictions,
- Metrics,
- Notes,
- Requirements,
- Reviewer's names, completion dates, comments,
- Sizing and timing estimates,
- Software problem reports,
- Tool output, such as a CASE tool, a compiler, or a PDL processor.

5.0 Analysis of DoD-Std-2167A SDF requirements

5.1 The total requirements

The total requirements have been determined by starting with the English language of DoD-Std-2167A and then breaking the constituent parts into elemental units. The English statements have compound objects expressed in a series or parallel form; thus, any particular sentence normally breaks into several elemental requirements. The standard follows the normal convention of indicating mandatory items by the verbs "shall" and "will"; the verbs "should" or "may" show optional requirements. The analysis of this section considers only mandatory requirements.

Table 5-1 shows the expansion of requirements that occurs from this analysis. The analysis used a derived definition of data-elements (presented later in this section) to determine the requirements elements.

5.2 Organization of the requirements

An organization of the SDF requirements is shown in Figure 5-1. At the first level the requirements are divided into *functional* and *constraints* (non-functional); at the second level the division is *structure* and *action*. Structure refers not only to what data elements must be stored, but also to what software components the elements are associated (CSCI, CSC, and CSU). The third level, which occurs only under action, has three divisions, *store*, *update*, and *evaluate*. Store refers to the storage of the information for each data element, update refers to changing the information when required, and evaluate refers to the extensive evaluation criteria described in Figures 7, 8, 9, and Appendix D of DoD-Std-2167A.

5.3 SDF data items

The elemental data items required in SDFs are not simple numbers, dates, times or other

numeric data. The data items imply a mix of text and numeric data that is characteristic of the informal origins of SDFs. The following list contains the data items:

1. Design considerations
2. Design constraints
3. Design documentation
4. Design data
5. Schedule information
6. Status information
7. Test requirements
8. Test responsibilities
9. Test schedules
10. Test cases (inputs, expected results, evaluation criteria)
11. Test procedures
12. Test results
13. Limits of specified requirements
14. Specified percentage of...SDFs

5.4 Timing during the life cycle

Figure 5-2 shows when the various requirements apply to a software development during the DoD-Std-2167A life cycle. The structure related requirements occur first and are defined in the developer's SDP, which is a formal deliverable under DoD-Std-2167A. The remaining requirements for storing, updating, and evaluating the contents of the SDFs occur during the DoD-Std-2167A software development process.

5.5 Critique of DoD-Std-2167A SDF requirements

Most of the DoD-Std-2167A requirements for SDFs are reasonable; however, the requirements for "evaluation" are unreasonable. The requirements in this category are from the extensive evaluation criteria described in Figures 7, 8, 9, and Appendix D of DoD-Std-2167A. The difficulty arises because the SDFs are informal working level documents and not the formal documents called for in a DID. Therefore, it is not good practice to apply the evaluation criteria of Appendix D to SDFs. Simpler more concise criteria should be applied to SDFs.

High level requirement #	Sentences	SDF requirement elements
R1	1	3
R2	8	27
R3	1	1
R4	2	7
R5	3	7
R6/R7	12	78
R8	1	1
R9	2	3
R10	1	1
R11	1	1
R12/R13	10	35
R14	1	2
R15	1	3
R16	1	1
R17/R19	6	38
R18/R19	6	38
R20	1	3
R21	2	6
21	60	255

Table 5-1 Requirements Expansion

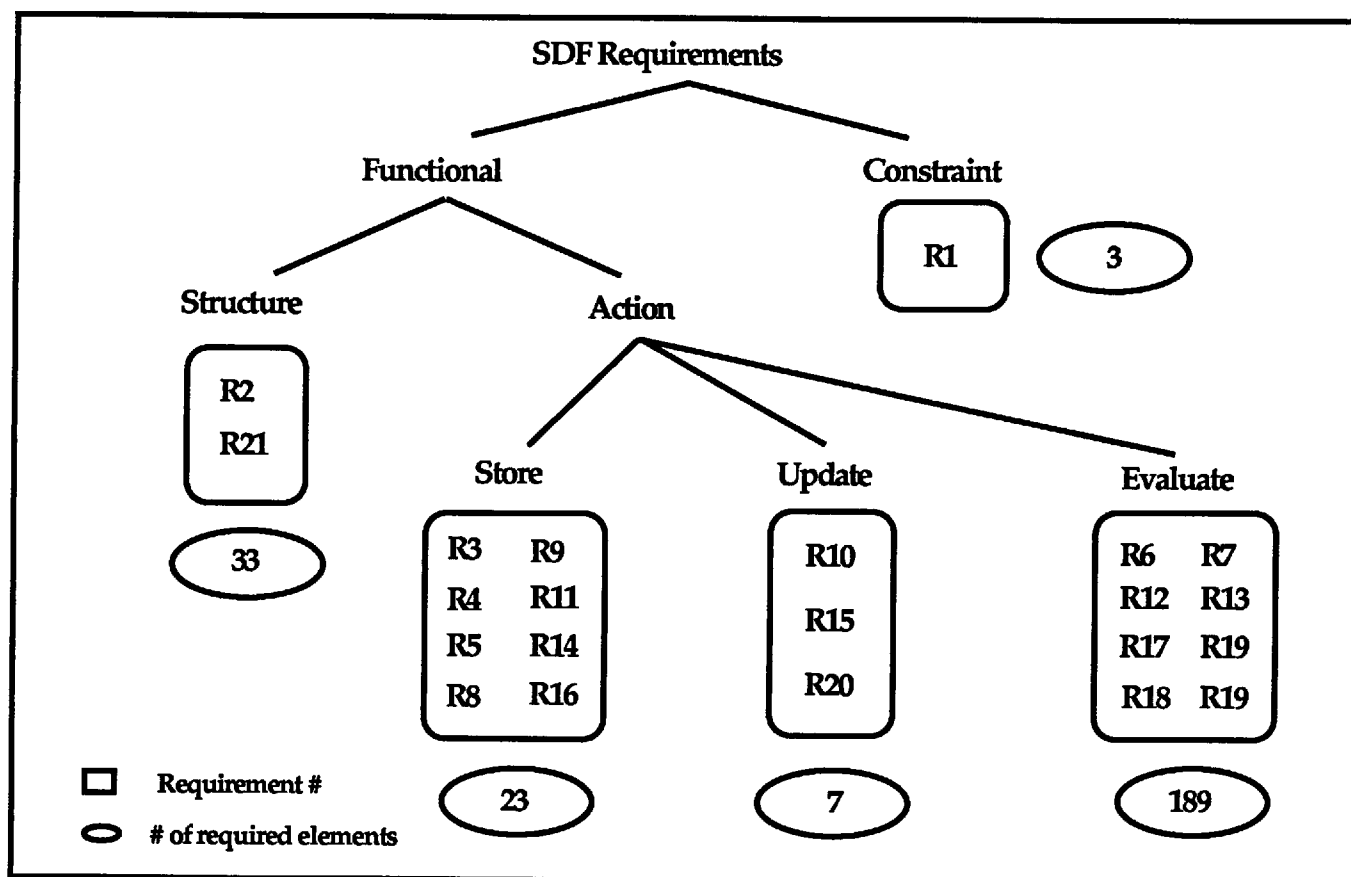


Figure 5-1 SDF Requirements

Even if the problems with the evaluation criteria are corrected, the full implementation of the remaining DoD-Std-2167A requirements would demand a considerable effort. It is probably not cost effective to apply these remaining requirements to small or modest sized software developments. Unfortunately, although current policy strongly encourages acquisition streamlining, there are no guidelines in DoD-Std-2167A or DoD-Hbdk-287 regarding the sizing of software developments and the SDF requirements.

6.0 Recommended process for SDFs

6.1 The SDF process

The SDF process recommended by this Guide is shown in Figure 6-1. The process generally follows the form of DoD-Hbdk-287 but applies specifically to SDFs. The first two steps occur before the contract or sub-contract; these steps are performed by either the government or the prime contractor. The remaining steps are performed by either the prime contractor or a subcontractor.

6.1.1 Step 1..Establish project DoD-Std-2167A tailoring

The considerations pertinent to SDFs interact with one another so a careful analysis is required. In general, SDFs will require an initial investment to establish the procedures and mechanisms for recording the basic information. Further, during the conduct of the software development there will be additional recurring expense (from a combination of human and computer resources) to actively use the SDFs. If the SDFs are implemented in a highly automated computerized form then the initial investment will be high but the recurring costs will be low. The costs of SDFs must be examined in consideration of their benefits; the benefits are improved status reporting of the software during development and potentially reduced post deployment software support costs.

The key considerations are:

current system life cycle phase,
software criticality
new development vs. modification,
deliverable and non-deliverable software,
cost, schedule, and reliability constraints,
externally-imposed requirements and policies,
design method,
size and complexity of software,
acquisition strategy,
support concept,
management approach.

6.1.2 Step 2 Tailor DoD-Std-2167A SDF requirements in the contract statement-of-work

Tailoring is the process of evaluating each requirement in a standard or DID to determine if it is suitable and cost-effective for a given project. For standards, such as DoD-Std-2167A, a modified version of any requirement may be included in the contract Statement of Work (SOW). For DIDs, requirements may be deleted or partially deleted, but not modified.

Except for the SDP (see R21 on page 4), SDFs are not specified in any DID; therefore, virtually all SDF requirements that differ from DoD-Std-2167A will be stated in the SOW. For example, the extensive SDF product evaluations required by DoD-Std-2167A may be deleted or reduced by appropriate wording in the SOW. Such changes need to be carefully made since if one quickly decided to drop Appendix D to eliminate SDF product evaluations they would later discover the product evaluations were also dropped for the project's formal documentation.

There are two aids to assist the tailoring of DoD-Std-2167A. One is DoD-Hbdk-287 A Tailoring Guide for DoD-Std-2167A, the second is TAILOR/2167A (a registered trademark of Logicon, Inc.), a commercially available computer program.

DoD-Hbdk-287 A Tailoring Guide for DoD-Std-2167A, provides guidance to Govern

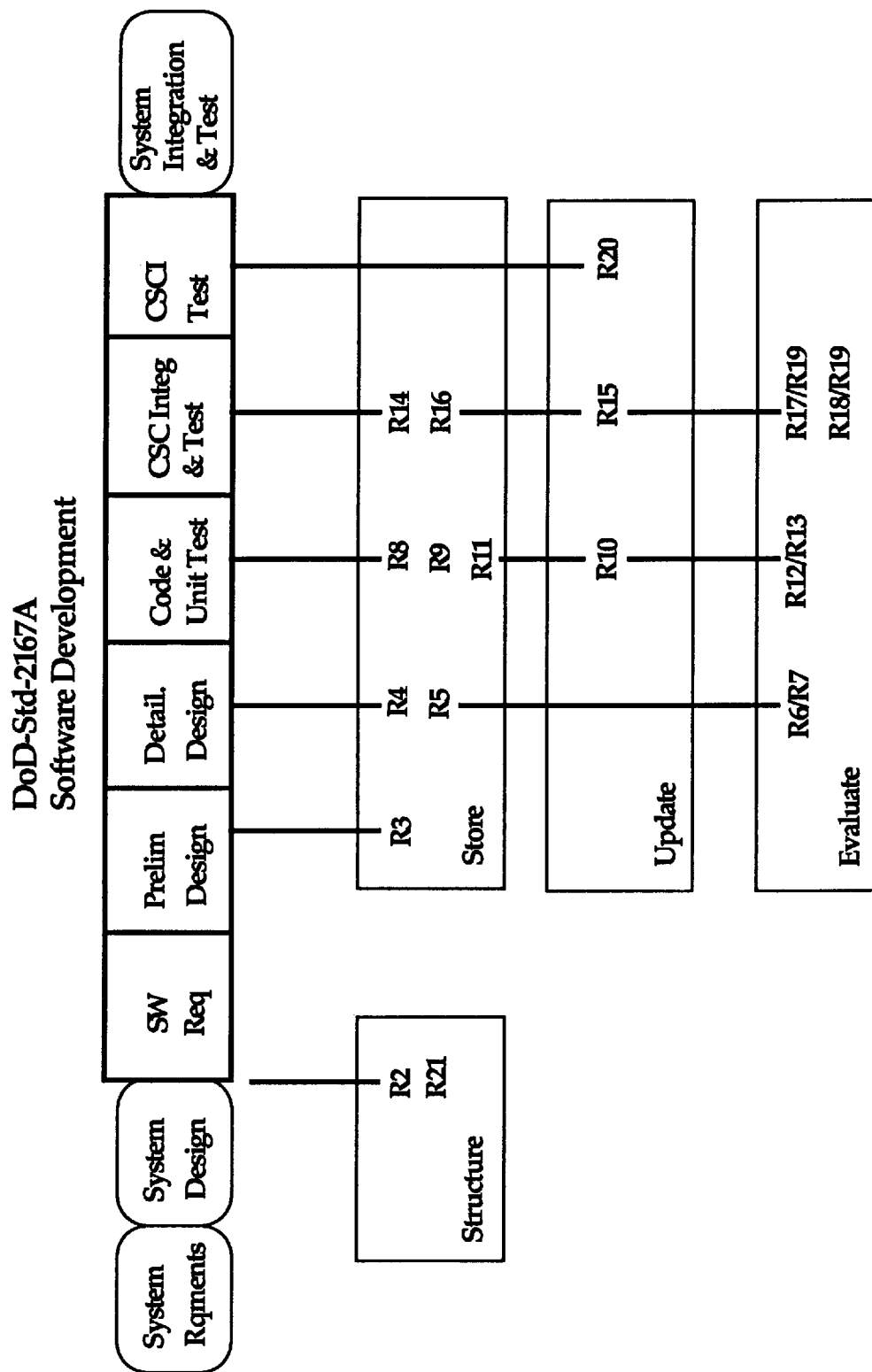


Figure 5-2 Timing of SDF Requirements During the DoD-STD-2167A Lifecycle

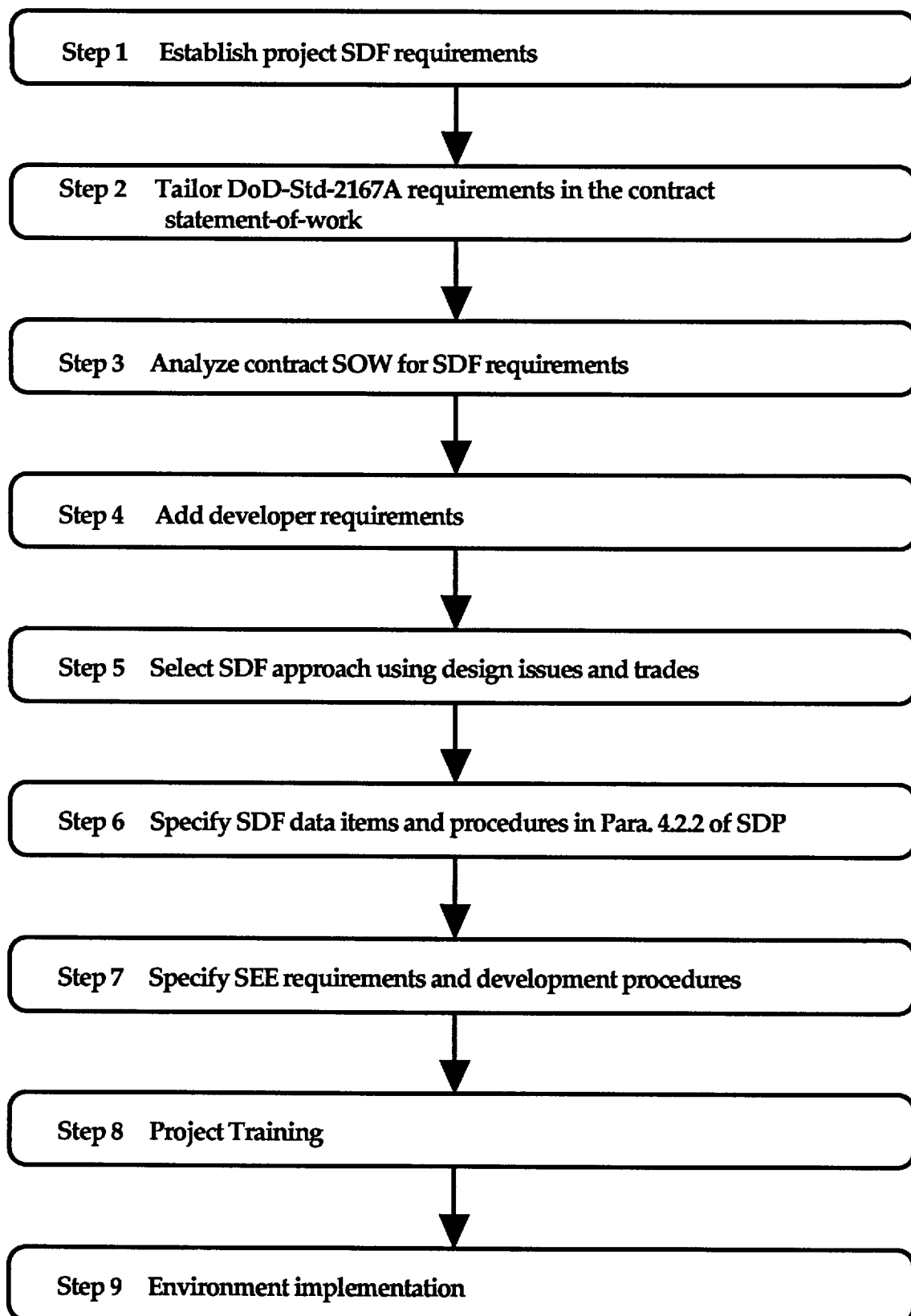


Figure 6-1 SDF Process

ment program managers and other program office staff that are responsible for tailoring DoD-Std-2167A for a software development or support contract. It explains key concepts and tailoring issues and describes how to tailor the standard and the associated DIDs. The handbook includes examples for each of the major life cycle phases. The use of DoD-Hbdk-287 is recommended.

TAILOR/2167A is a commercially available computer tool to assist the tailoring of DoD-Std-2167A requirements. This tool is available free of charge to authorized DOD users under a licensing agreement with the Joint Logistics Commanders Joint Policy Group on Computer Resource Management. The use of TAILOR/2167A is recommended but not required.

An examination of Version 2.0 of TAILOR/2167A showed at least one serious omission regarding SDFs. The TAILOR/2167A program has a feature that shows the DoD-Std-2167A requirements grouped by topic. The SDFs are listed as a topic and when selected all applicable paragraphs are supposed to be listed; however, paragraph 5.5.2.2 of DoD-Std-2167A does not appear on the list. Further, in TAILOR/2167A the evaluation criteria is not listed with the SDFs but under two separate topics, making it difficult to correlate with SDF requirements.

A good feature of TAILOR/2167A is the ability of the user to request advice, which is pre-stored information on a particular paragraph of DoD-Std-2167A. The information provided by TAILOR/2167A is generally considered good advice and should encourage the user to thoughtfully and thoroughly tailor all DoD-Std-2167A requirements.

6.1.3 Step 3 Analyze contract SOW for SDF requirements

A contractor or software developer must thoroughly analyze the SOW for SDF requirements. The best case is when such issues are resolved before contract; Changes after award are usually time consuming.

A software developer can start with the SDF

requirements shown in paragraph 4.1 and then delete or modify them depending on the requirements in the SOW. The result should be a firm set of SDF requirements from the procuring activity.

6.1.4 Step 4 Add developer requirements

The software developer may then add SDF requirements that are necessary for his organization's methods and procedures. Typical items are shown in paragraph 4.2.

6.1.5 Step 5 Select SDF approach using design issues and trades

A software developer, who does not have an automated SDF capability but intends to build one, should refer to the design issues and trades described in paragraph 7.

6.1.6 Step 6 Specify SDF data items and procedures in Para 4.2.2 of SDP

With the full requirements and an implementation approach defined, the software developer can complete the parts of the SDP that apply to SDFs. Once it is approved, the SDP becomes the major control document for not only SDFs but also the entire project's software development.

6.1.7 Step 7 See specifications and development procedures

With an approved SDP, the software developer can specify the detailed SEE requirements and the operational procedures that will help the individual software engineers to properly create, use, and maintain the SDFs.

6.1.8 Step 8 Project training

This step covers the development of training materials and the training of the software engineers in the methods and procedures necessary for the creation, use, and maintenance of SDFs.

6.1.9 Step 9 Environment implementation

This step covers the implementation or modification of the software developers SEE to automatically implement SDFs; this step also covers the testing of all SDF related software.

7.0 Design issues and trades

7.1 General issues for developing the SDFs

An SDF can be created manually on paper, electronically using a file directory system, or electronically using a data base. These three approaches can be used individually or in combination with each other. A project must decide what approach is best for its particular application. If the job is small, it may be more cost-effective to produce the SDFs manually on paper. However, if the job is large, the SDFs can be created by using a file directory structure or a data base with minimal manual interaction.

When selecting the best approach for the project, the type of software tools available on the host computer system must be considered. The following functions will be needed:

- data collection,
- data storage,
- data access,
- data retrieval,
- data security,
- creating a directory structure,
- attaching to a data base.

Creating a flexible structure may be essential since there is not necessarily a one-to-one correspondence between SDFs and CSCs or CSUs; DoD-Std-2167A permits one SDF for a set of logically related CSCs or CSUs.

Several alternative SDF implementations should be investigated and then compared in cost, schedule, and impact on the primary software development effort. The SDF implementation should be planned with the same thoroughness as any important software

effort. Testing and integration are particularly important since they must be completed before significant use is made of the SEE.

7.2 SDF data flow analysis and design

An excellent initial step is to do data analysis, develop data structures, and construct data flows for information into and out of the SDF. The SDF related processing is the same as commercial transaction data processing; there are numerous commercially available methods and software tools that can help with this activity. Important issues are the computer resource capabilities, the centralization of data, and whether the data is distributed over more than one system.

A fundamental issue is whether all SDF information should be in separate SDF files in a separate SDF directory or if it should be distributed throughout the software development files. A consideration is whether the software developers are all working on the same computer system. The accessibility of the data should be considered when deciding where and how the data will be stored. For example, if naming conventions and key words are used they will have to comply with various tool interfaces.

An additional consideration is the tools that are needed to both create and use the SDFs. These might include tools for searching, tools for extracting SDF data from source files or SDF files, and tools for creating SDF reports or other outputs. The output from tools normally used in the SEE is a potential source of SDF information. A way of capturing and storing this information in the SDF must be devised. This output includes cross reference information, PDL (Program Design Language) reports generated by software development tools, and compiler output.

7.3 Organizational structure and roles

The project organizational structure and roles of the team members are important items for

any software development project. The roles of the people within the organization, as well as their view of the SDF, must be defined. These people include managers, designers, programmers, testers, reviewers, and QA and CM personnel. For example, Who is responsible for the SDFs?; Who has write-access to the SDFs?; Who has the authority to change the SDFs to read-only? Another issue is how the communication between team members takes place. For example, do people interact through a computer system, by paper, or face-to-face?

7.4 Storage of SDF information

A major issue of concern to the developer and maintainers of the SDF is the storage of SDF information. There are several issues concerning the storage of SDF information:

- Is the SDF distributed over different systems?
- Is the SDF information redundant?
- Is the SDF information distributed across networks?
- Is the SDF information distributed across non-communicative systems?
- Is the stored data disjoint?

7.5 Use of a file directory structure

If a directory file structure is selected, the following issues are important:

- file capacity per directory,
- the maximum levels allowed in the directory structure,
- file naming conventions,
- file layout,
- data accessibility,
- data storage mechanism,
- data security,
- localizing all SDF information,
- distributing SDF information throughout the software development files,
- redundancy of data.

7.6 Limitations on file naming conventions

If a file directory structure is used, the file naming conventions must be considered. The naming conventions must be consistent with the SDF and the project software naming conventions. Consistency is important, especially if special tools are being used to extract SDF information from the source files as well as the SDF files.

7.7 Use of a data base management system (DBMS)

If a data base is used for developing the SDFs on a software development project, the following issues should be considered:

- the type of data model for creating the SDFs
- the type of DBMS for creating the SDFs
- the capabilities of the DBMS
- the limitations in using the DBMS
- the ease of use of the DBMS
- the methods used for handling data security
- the configuration management capabilities of the DBMS.

7.8 Special SDF tools

If special tools are to be used for storing and extracting data from the SDF files, then a general file layout should be developed, including a list of standard key words to use for ease of access. If no tools are available on the computer system to aid the development of the SDF, the project may elect to develop their own set of tools.

If the data is centralized on one system, the tools may be developed so there is one common user interface to all SDF tools. If the data is distributed over different computer systems, the tools that are available on each system and the ability to access information between systems must be considered.

There are several additional issues concerning software development environment tools and the development of the SDF:

- How easy can the data be accessed?
- How is SDF information within different software environments referenced?
- How is the communication flow between the people in the organization accomplished?
- What computer resources are available?
- What is the organizational structure?
- How are problems resolved?

7.9 Configuration management of SDF information

There are two important issues related to configuration management of SDF information: controlling changes to SDF files during development and tracking SDF status that is associated with the various releases. Related issues include the method and duration of archive information.

7.10 Security

There are several security issues that will influence the collection, storage, and accessibility of the SDF information:

- security impact
- priority data
- test
- distributed storage of SDF information
- test results
- design constraints
- secure environment (if totally secure, no impact)
- mixed environment (secure and non-secure SDF).

8.0 References:

Boehm, Barry W. *Software Engineering Economics*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey 07632, pages 607-612.

Distaso, J.R. "Software Management--A Survey of the Practice in 1980", TRW Software Series, No. TRW-SS-80-10, September 1980, Redondo Beach, California 90278, pages 13-14.

Williams, R.D. "Managing the Development of Reliable Software," Proceedings, 1975 International Conference of Reliable Software, IEEE/ACM, April 1975, pages 3-8.

Ingrassia, F.S. "Combating the 90% Syndrome," *DATAMATION*, January 1978, pages 171-176.

9.0 List of acronyms and abbreviations:

CASE	Computer Aided Software Engineering
CM	Configuration Management
CSC	Computer Software Component
CSCI	Computer Software Configuration Item
CSU	Computer Software Unit
DBMS	Data Base Management System
DETAIL.	Detailed
FQT	Formal Qualification Testing
INTEG	Integration
NDS	Non-Developmental Software
PDL	Program Design Language
PRELIM	Preliminary
QA	Quality Assurance
REQ	Requirement
SDD	Software Design Document
SDF	Software Development File
SDL	Software Development Library
SDP	Software Development Plan
SEE	Software Engineering Environment
SOW	Statement of Work
STD	Software Test Description
UDF	Unit Development Folder

American Institute of Aeronautics and Astronautics

**The Aerospace Center
370 L'Enfant Promenade, S.W.
Washington, D.C. 20024-2518**

ISBN 0-930403-81-9